Tampereen yliopisto
Tampere University

TIE-02306 ”ItSE”

Introduction to Software Engineering

5 credit units

05-diagrams-ItSE-2019-v4

Tampereen yliopisto
Tampere University

Course contents (plan)

1. Course basics, intro
2. Sw Eng in general, overview
3. Requirements
4. **Basic UML Diagrams (”Class”, Use Case, Navigation)**
5. UML diagrams, in more detail
6. Different software systems
7. Life Cycle models
8. Quality and Testing
9. Project work
10. Project management
11. Open source, APIs, IPR
12. Embedded systems
13. Recap

Tampereen yliopisto
Tampere University

5. UML diagrams, in more detail

- Just a few more UML diagrams…
  - **state machine diagrams** (”state transition diagrams”, ””state diagrams, ”state machines”)
  - **sequence diagrams** (”scenarios”)
  - **activity diagrams**
  - activity diagrams with partitions (”swimlane diagrams”)

Tampereen yliopisto
Tampere University

Current at course (w 40)

- we have eight project groups left
- **WE5 groups this week are on WED and THU**
- after those we think next week WE6 groups
- **to WE5 BYOC (bring your own computers); Dia tool**
- **Invite tunitensu to your group’s Trello board**
- ***Registration for EXAM 1/3 (weeks 41-43) is open***
- **1st phase return/delivery section is at Moodle (opens 06.10.)**

Remember InnoEvent (04-08.11.2019) www.innoevent.fi

Backlog items with deadline

- **09.09.2019** at 23:59 Group forming (Moodle)
- **15.09.2019** at 23:59 Trello creation (Trello)
- **13.10.2019** at 23:59 Phase 1 documentation (Moodle)
- **13.10.2019** at 23:59 Phase 1 presentation slides (PRP-tool)
- **Week 43** Phase 1 presentations (Physical realm)
- **03.11.2019** at 23:59 Phase 1 peer feedback (PRP-tool)
- **17.11.2019** at 23:59 Phase 2 documentation (Moodle)
- **17.11.2019** at 23:59 Phase 2 presentation slides (PRP-tool)
- **Week 47** Phase 2 presentation (Physical realm)
- **01.12.2019** at 23:59 Phase 2 peer feedback (PRP-tool)
- **08.12.2019** at 23:59 Final delivery of project documentation (Moodle)
- **15.12.2019** at 23:59 Final peer feedback and self assessments (PRP-tool).

Tampereen yliopisto
Tampere University

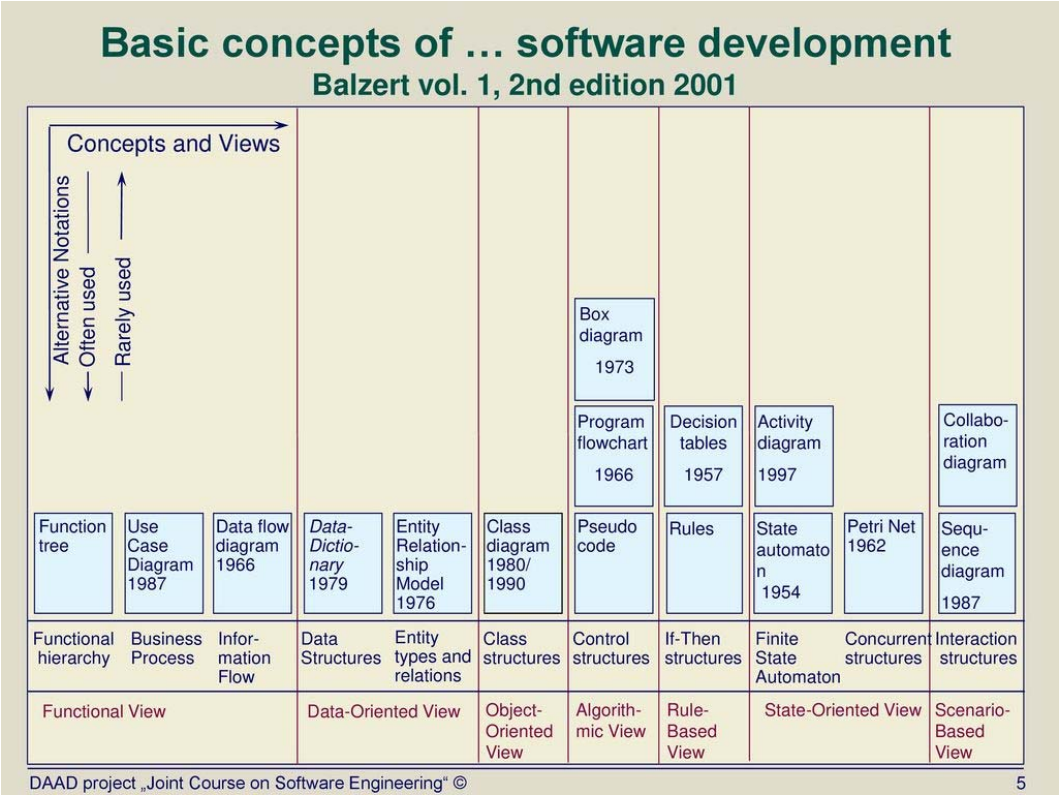
Weekly exercise attendees

	w36 WE1	w37 WE2	w38 WE3	w39 WE4	w40 WE5	w41 WE6	w44 WE7	w45 WE8	w46 WE9	w48 WE10
WED	0	14	9	5						
THU	21	13	14	17						

We will continue two Weekly Exercise groups, as long as the number of attendees are reasonable.

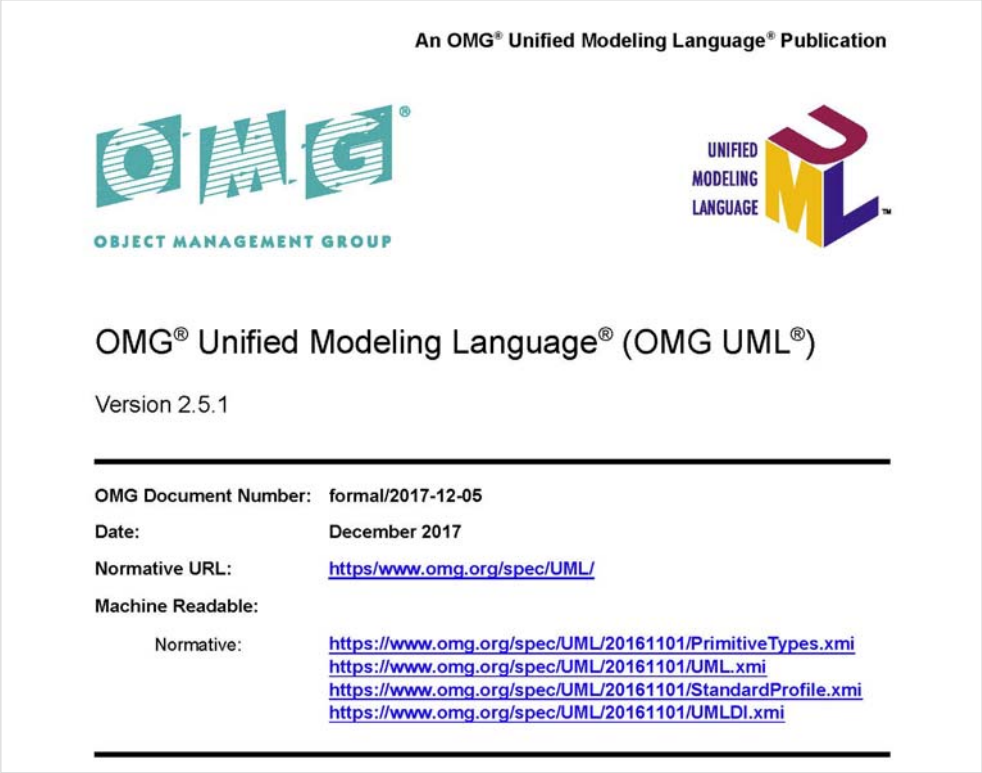
Tampereen yliopisto
Tampere University

Yes, there are many and many different kind of **methods and diagramming techniques**… for example classified by view, just pick the most useful to your specific project and use them.



UML 2.5.1 (2017) specification

www.omg.org/spec/UML/2.5.1/PDF



An OMG® Unified Modeling Language® Publication

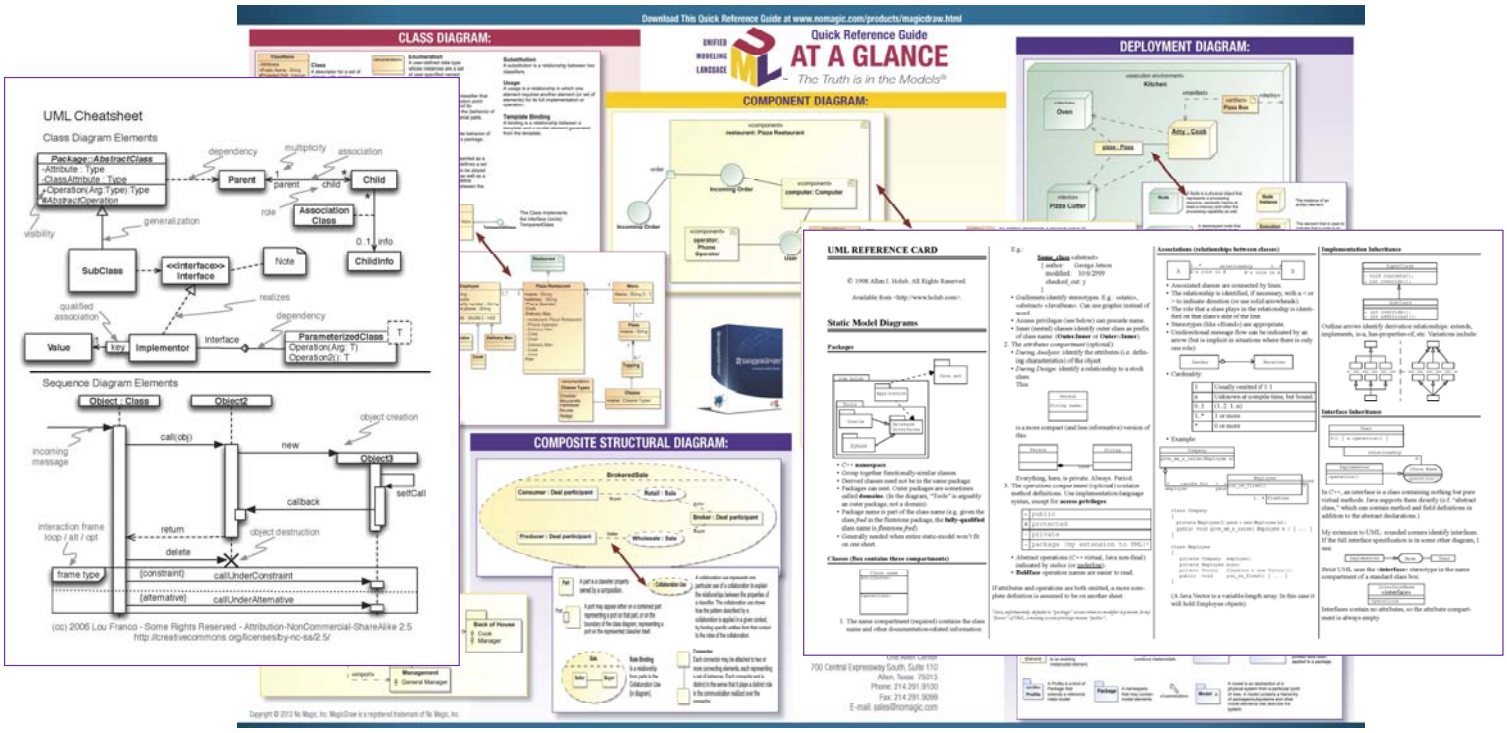
OMG® Unified Modeling Language® (OMG UML®)

Version 2.5.1

OMG Document Number:	formal/2017-12-05
Date:	December 2017
Normative URL:	https/www.omg.org/spec/UML/
Machine Readable:	
Normative:	https://www.omg.org/spec/UML/20161101/PrimitiveTypes.xmi https://www.omg.org/spec/UML/20161101/UML.xmi https://www.omg.org/spec/UML/20161101/StandardProfile.xmi https://www.omg.org/spec/UML/20161101/UMLDI.xmi

Tampereen yliopisto
Tampere University

There are many UML reference cards and cheat sheets available



UML 2.5.1 (2017) specification

www.omg.org/spec/UML/2.5.1/PDF

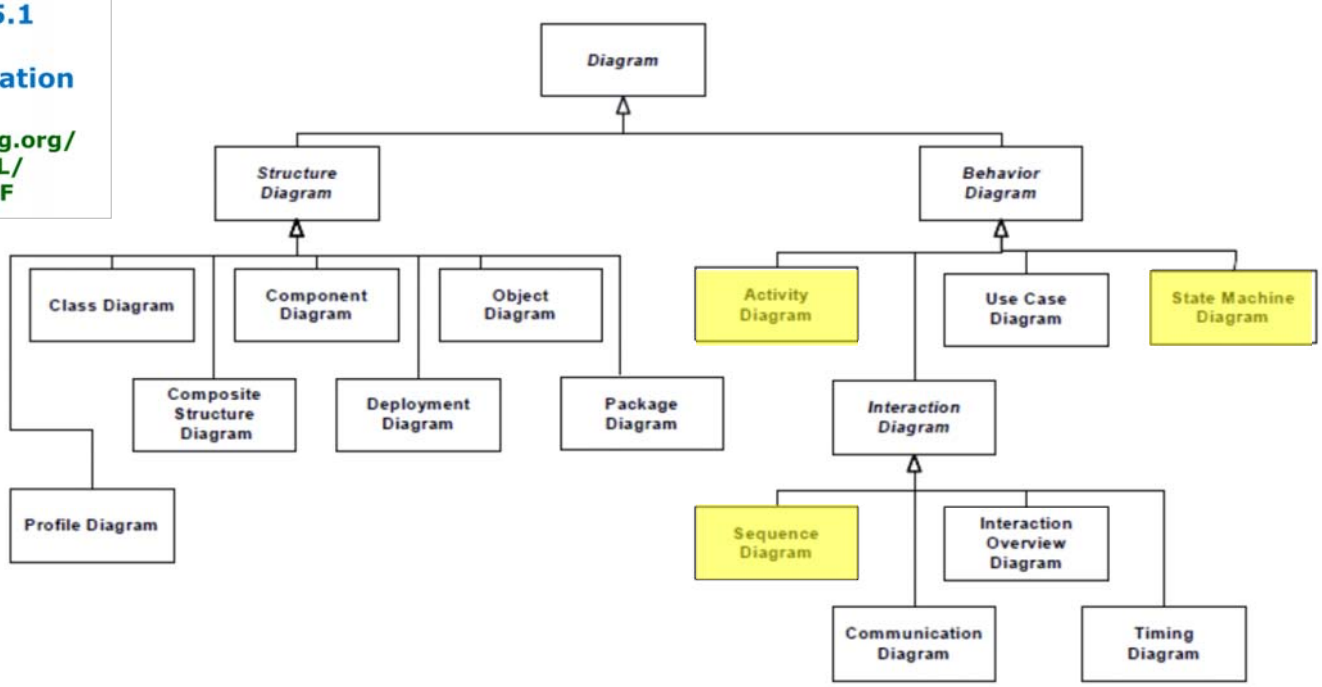
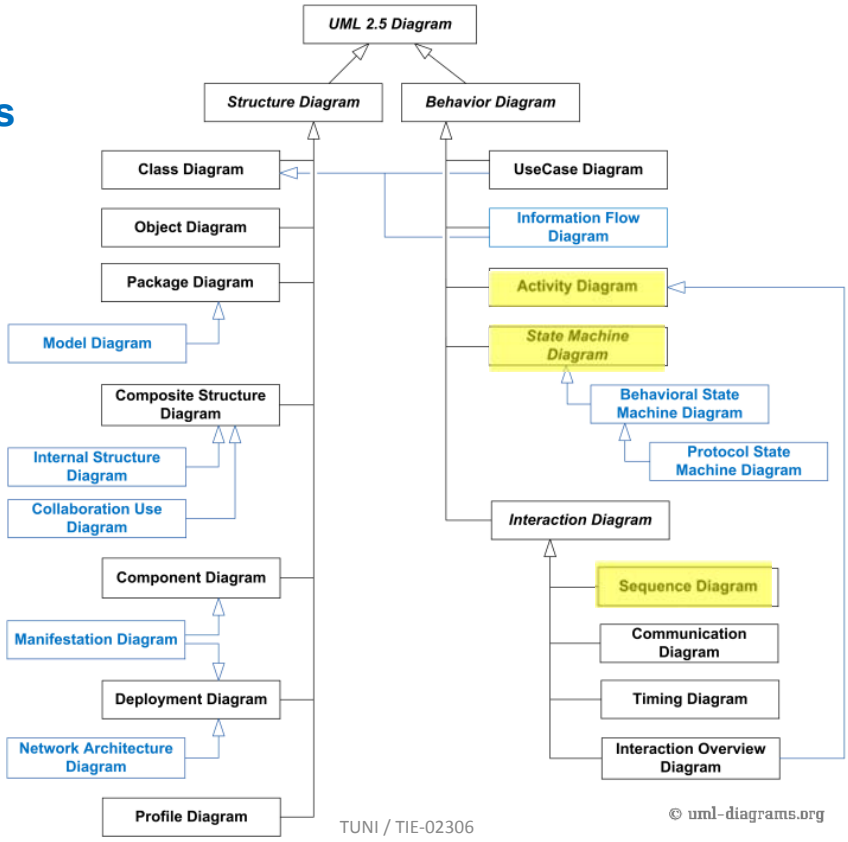


Figure A.5 The taxonomy of structure and behavior diagrams

UML 2.5 diagrams (2015)

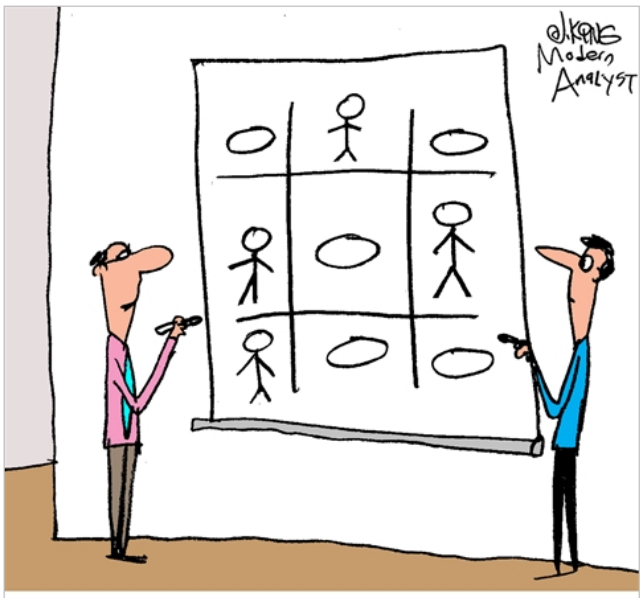
UML diagram types



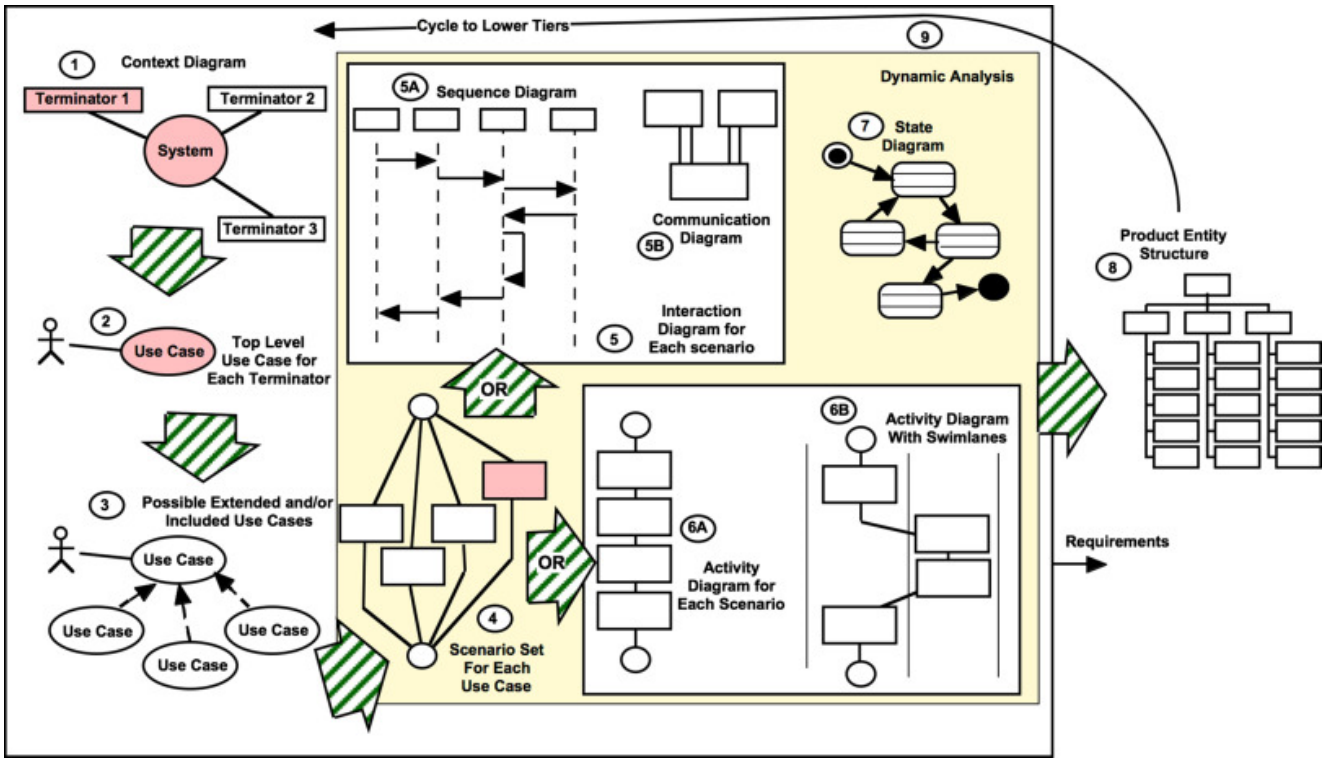
www.uml-diagrams.org/

Tampereen yliopisto
Tampere University

Feel free to use what ever kind of diagrams and drawings in your software engineering project. But remember to explain the symbols and techniques to the other parties.



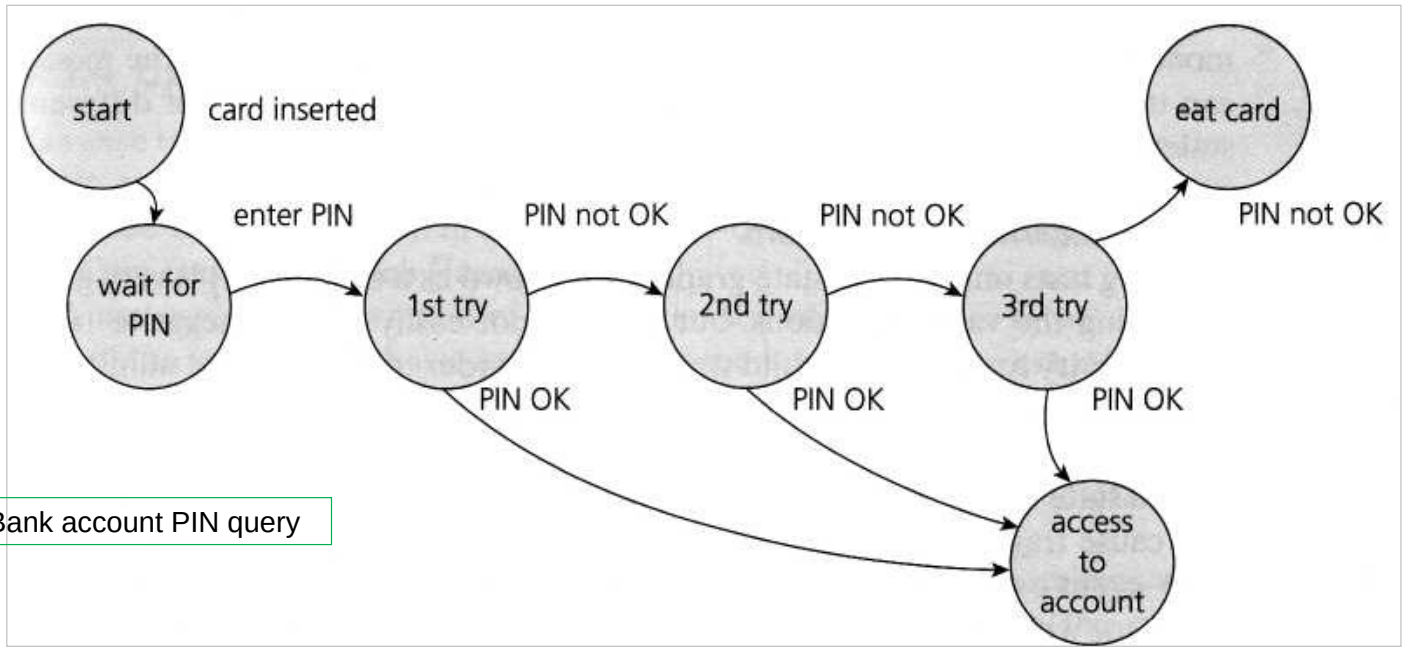
Jerry and Bob, the new business analysts, decided to settle, once for all, whether use cases or actors are more important in the use case diagram.



[ars.els-cdn.com]

State transition diagrams = State machines

Basic state diagram, may be used early in requirements specification phase



Bank account PIN query

[tryqa.com]

Use of state diagrams

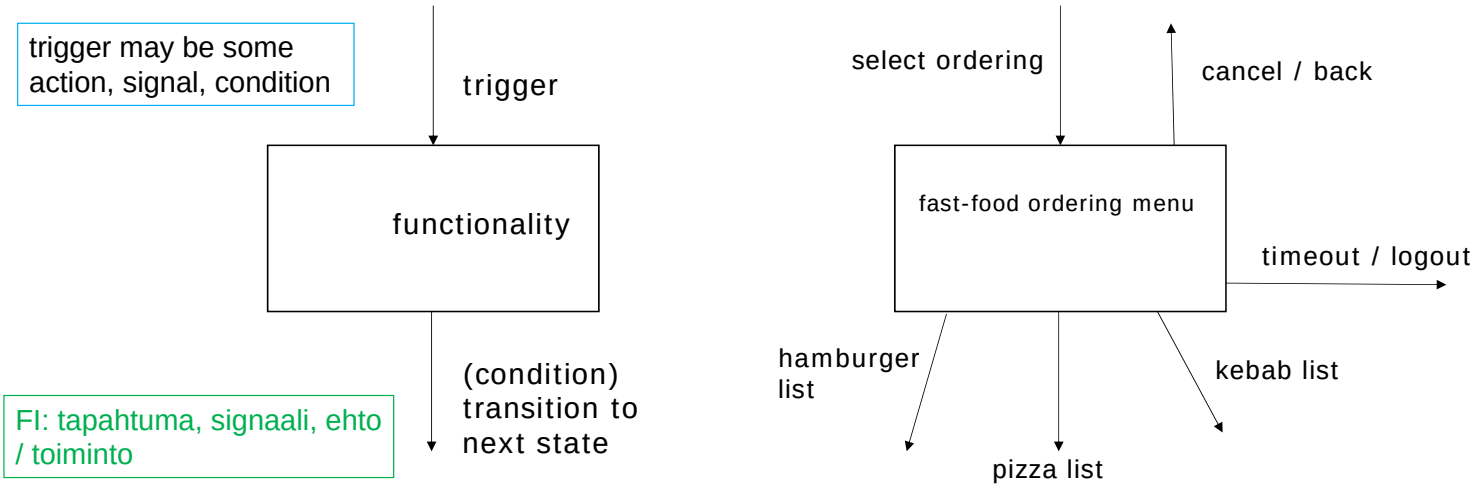
State diagrams may be used in ”general” way in requirements phase, and in more detail in design phase when defining software architecture and to help developers to understand software’s wanted behaviour in detail.

Sometimes state transition table (matrix) would help. From such it is easy to see all state transitions which are enabled. Sometimes it would be necessary in code **to make sure that certain state transitions do not happen**.

For example in some real-time systems it is very important in which order the functions are done.

State (transition) diagram, state machine (FI: tila(siirtymä)kaavio, tilakone)

Traditionally software's order of functionality (e.g. user's possible action paths) has been expressed by state transitions diagrams.



State machines [UML 2.5.1 spec]

The StateMachines package defines a set of concepts that can be used for modeling discrete event-driven Behaviors using a finite state-machine formalism. In addition to expressing the Behavior of parts of a system (e.g., the Behavior of Classifier instances), state machines can also be used to express the valid interaction sequences, called *protocols*, for parts of a system. These two kinds of StateMachines are referred to as *behavior state machines* and *protocol state machines* respectively.

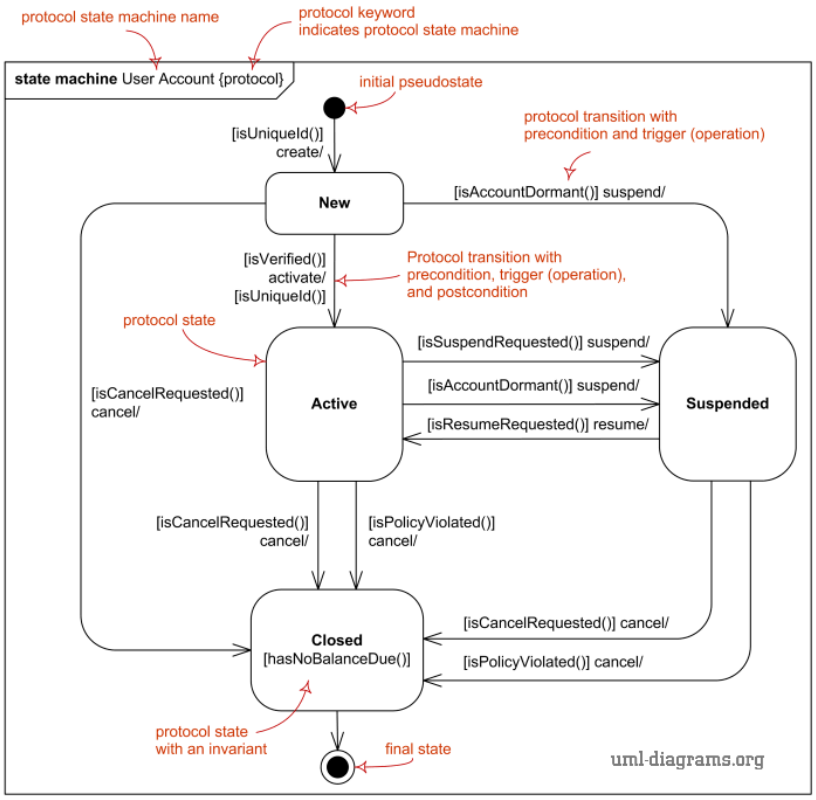
Behavior StateMachines can be used to specify any of the following:

- The classifierBehavior of an active Class.
- An ownedBehavior of a BehavioredClassifier that is not the classifierBehavior of that BehavioredClassifier.
- A stand-alone Behavior, that is, one that does not have a corresponding BehavioredClassifier.
- A method corresponding to a BehavioralFeature (i.e., an Operation or a Reception).

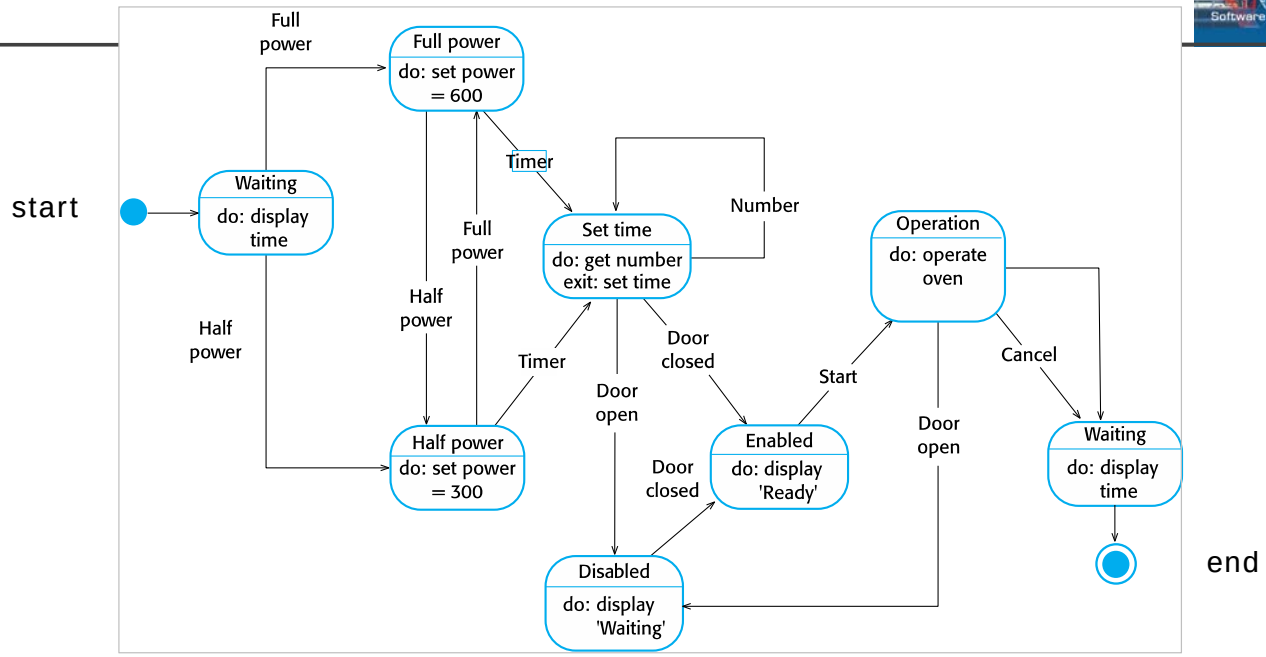
Tampereen yliopisto
Tampere University

UML protocol **state machine** diagrams are used to express a **usage protocol** or a **lifecycle** of some classifier. It shows which operations of the classifier may be called in each state of the classifier, under which specific conditions, and satisfying some optional postconditions after the classifier transitions to a target state.

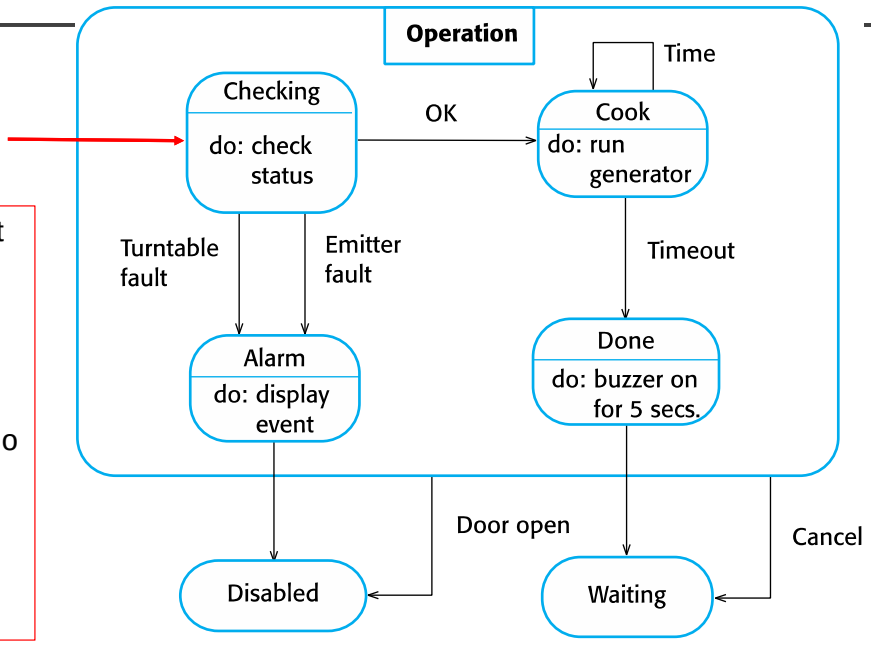
preconditon / postcondition



State diagram of a microwave oven



Microwave oven operation



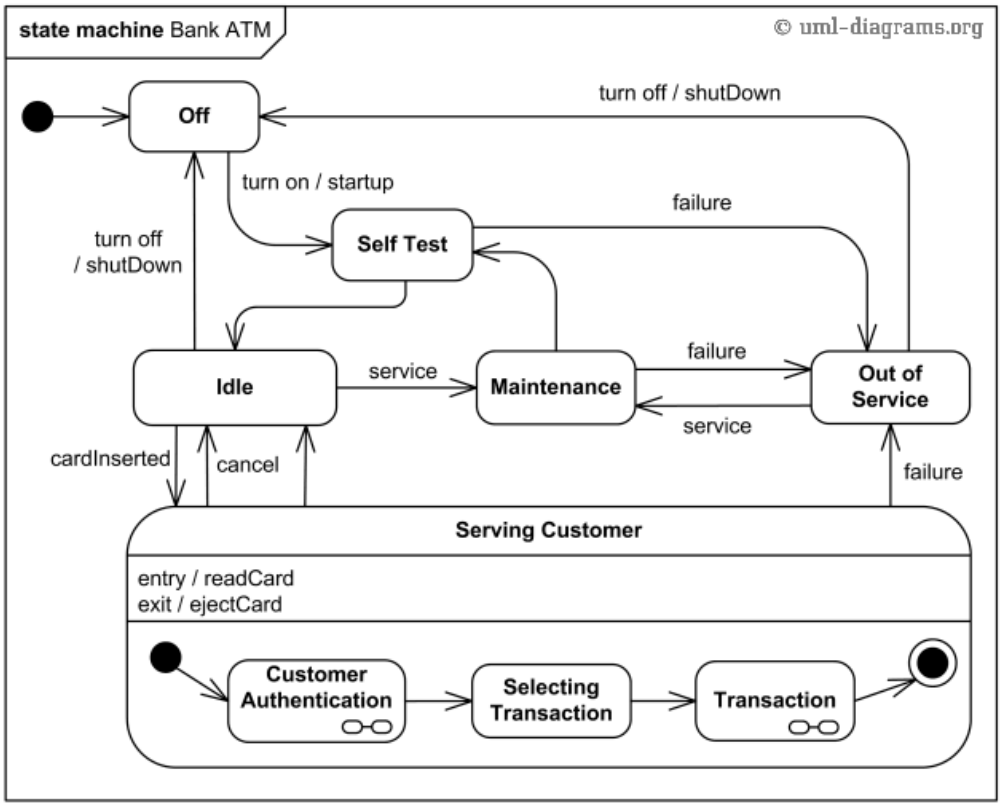
It is important to define from which state operating begins, when the device is turned on (also after power outage).
So it will not go to some random state.

States and stimuli for the microwave oven

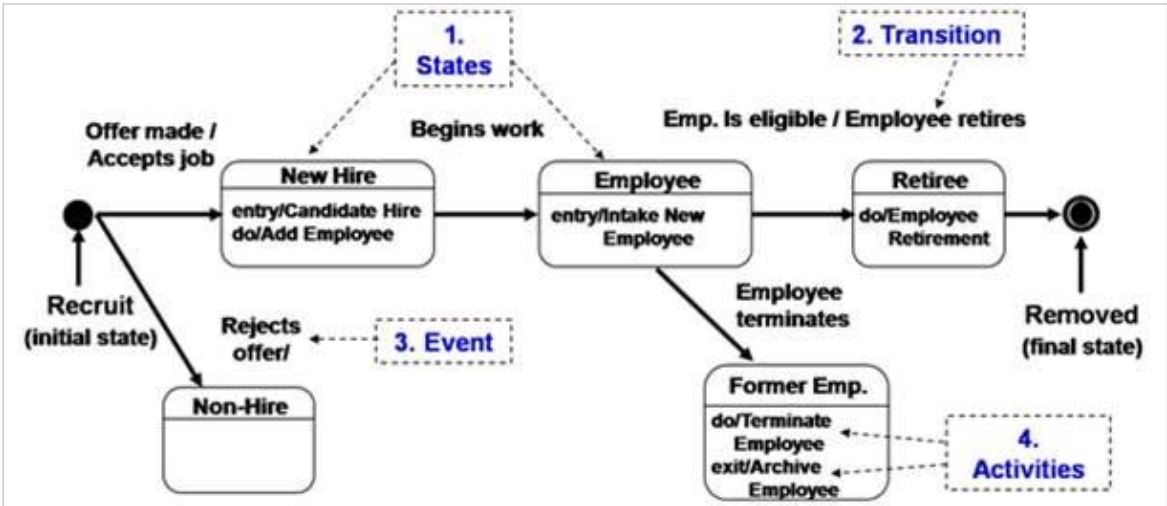
State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows 'Half power'.
Full power	The oven power is set to 600 watts. The display shows 'Full power'.
Set time	The cooking time is set to the user's input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows 'Not ready'.
Enabled	Oven operation is enabled. Interior oven light is off. Display shows 'Ready to cook'.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for five seconds. Oven light is on. Display shows 'Cooking complete' while buzzer is sounding.

Tampereen yliopisto
Tampere University

Bank Automated Teller Machine (ATM) top level state machine.



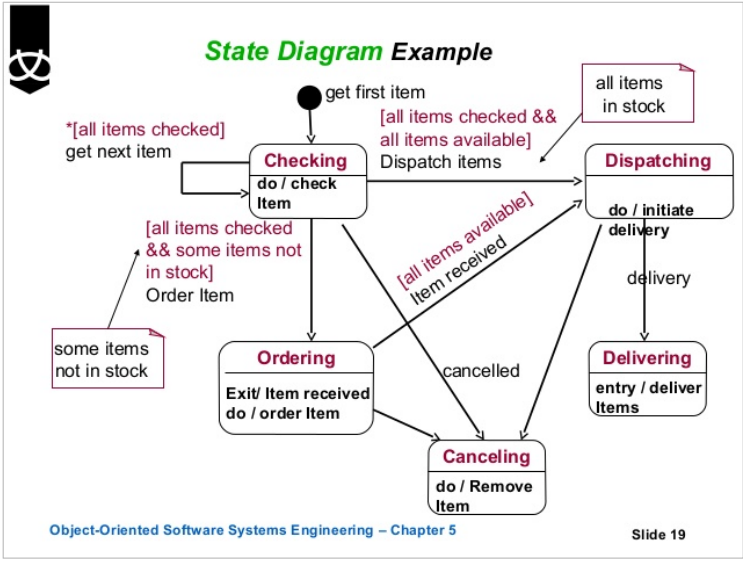
Tampereen yliopisto
Tampere University

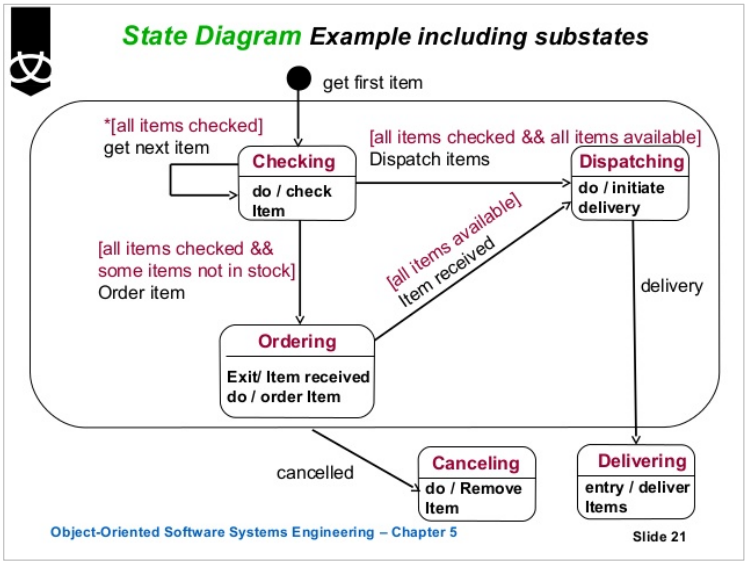


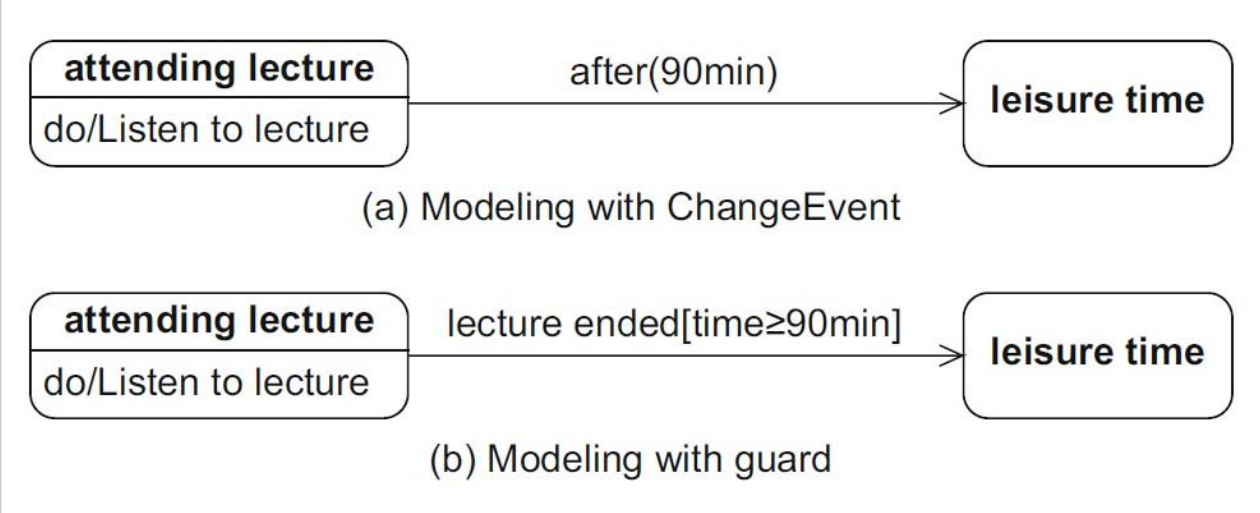
State diagram. It shows 1) Five example states for a Human Resources application, 2) Transitions that reflect the business process in moving from state to state, 3) The event or trigger that causes a move to a new state, and 4) Activities that can take place upon entering or exiting a given state.

[www.watermarklearning.com]

Two different diagrams (styles) of the same system, e.g. a web shop system







a) student has leisure time after 90 minutes at any case.
b) student has leisure time after lecture end, if the lecture has lasted at least 90 minutes.

A [guard] can never trigger an event itself.

[Figure 5.11, UML@Classroom, 2015]

Sequence diagrams

Sequence diagram lifelines

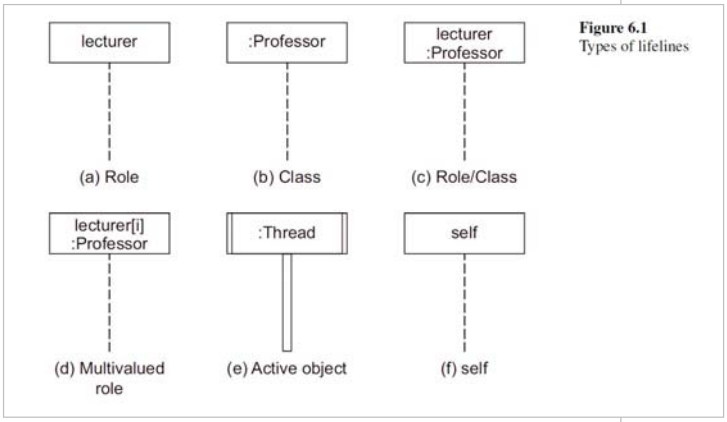


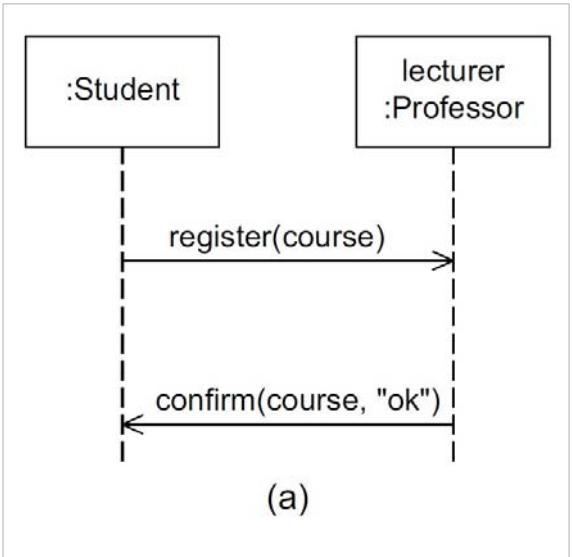
Figure 6.1
Types of lifelines

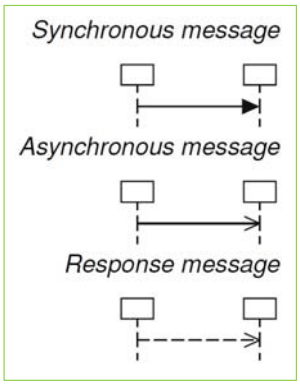
Table 6.2
Notation elements for the sequence diagram

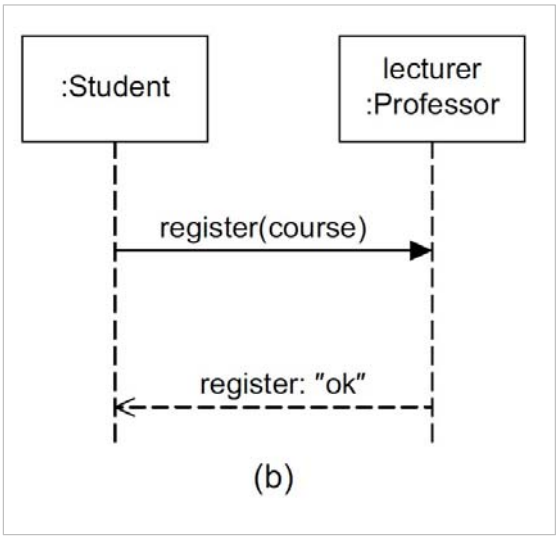
Name	*Notation*	*Description*
Lifeline	r:C A	Interaction partners involved in the communication
Destruction event		Time at which an interaction partner ceases to exist
Combined fragment	... [...]	Control constructs
Synchronous message		Sender waits for a response message
Response message		Response to a synchronous message
Asynchronous message		Sender continues its own work after sending the asynchronous message
Lost message	lost	Message to an unknown receiver
Found message	found	Message from an unknown sender

[UML@Classroom, 2015]

Sequence diagram, messages







[UML@Classroom, 2015]

Entity / Context diagram, or…

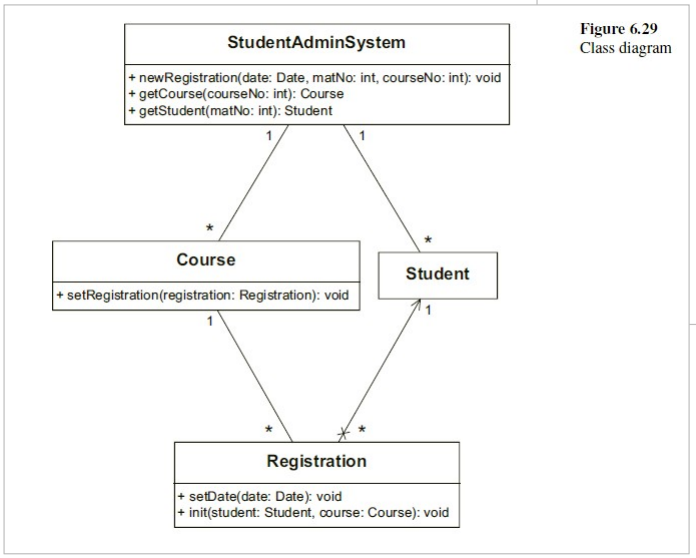


Figure 6.29
Class diagram

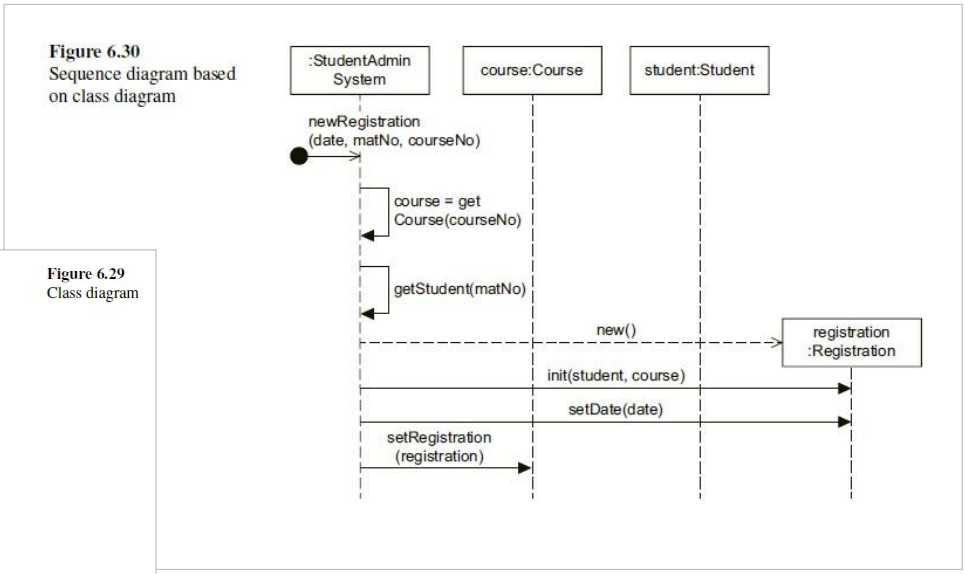


Figure 6.30
Sequence diagram based on class diagram

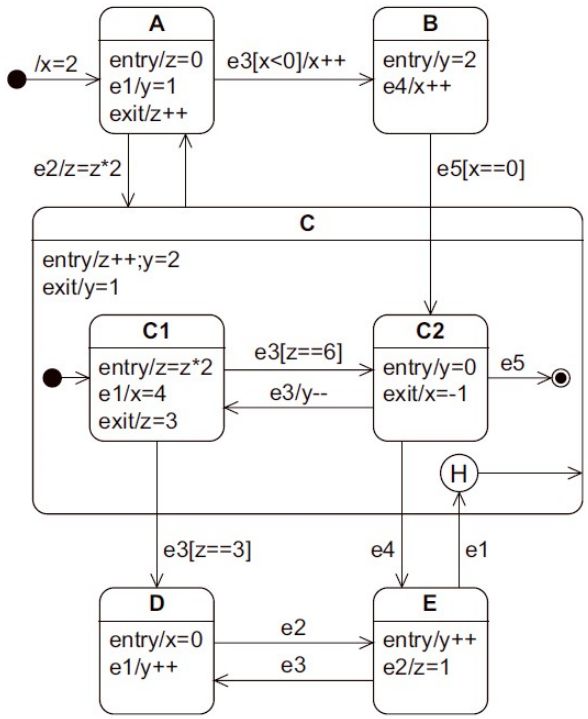
Sequence diagram

[UML@Classroom, 2015]

Tampereen yliopisto
Tampere University

State machine diagram

Figure 5.20
State machine diagram to demonstrate a sequence of events



One kind of state table

Table 5.1
State changes and variable assignments for *x*, *y*, and *z* after the occurrence of the individual events

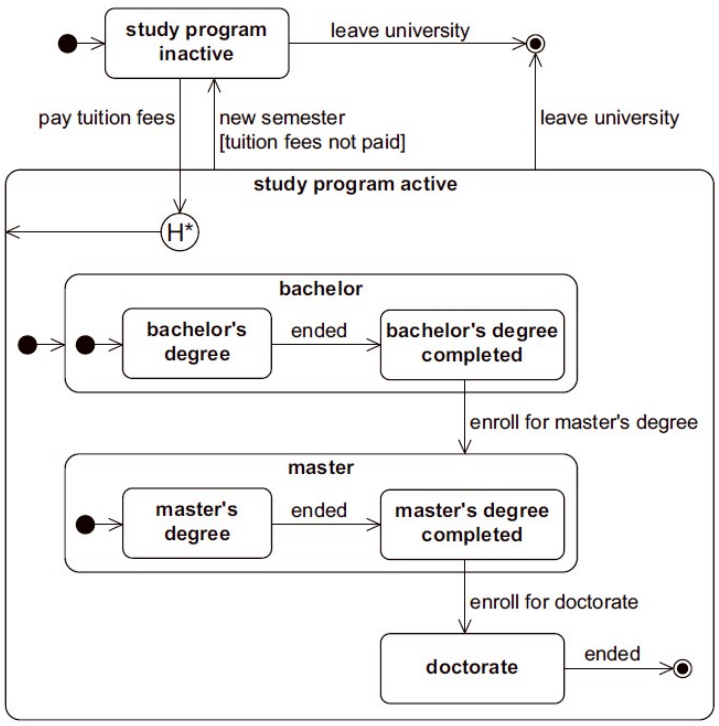
Event	State entered	*x*	*y*	*z*
Start	A	2		0
e2	C1		2	6
e1	C1	4		
e3	C2		0	3
e4	E	-1	2	
e1	C2		0	4
e5	A	-1	1	0

[UML@Classroom, 2015]

Tampereen yliopisto
Tampere University

State machine diagram

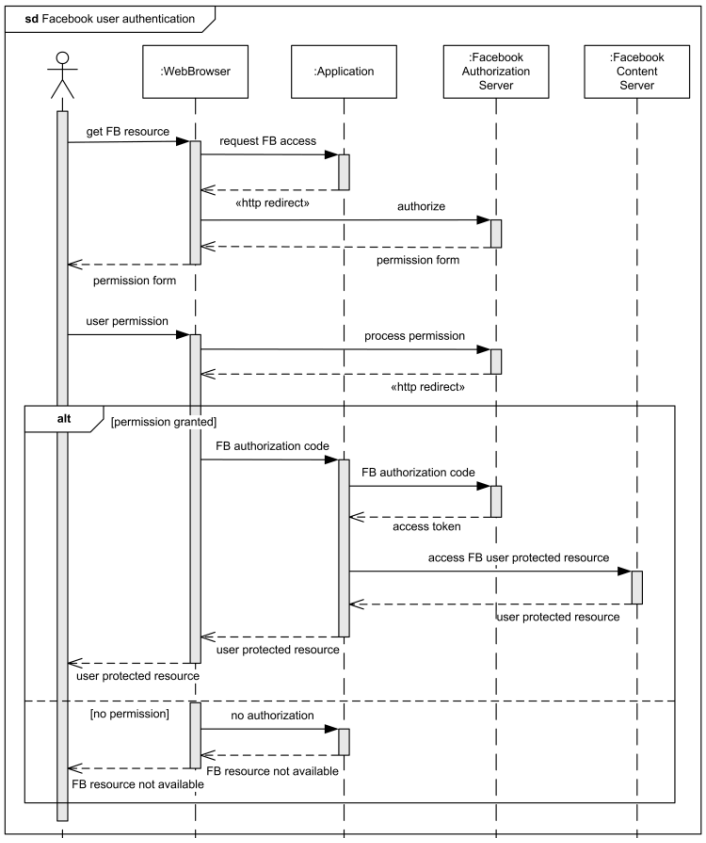
Figure 5.19
States of an academic education



[UML@Classroom, 2015]

Tampereen yliopisto
Tampere University

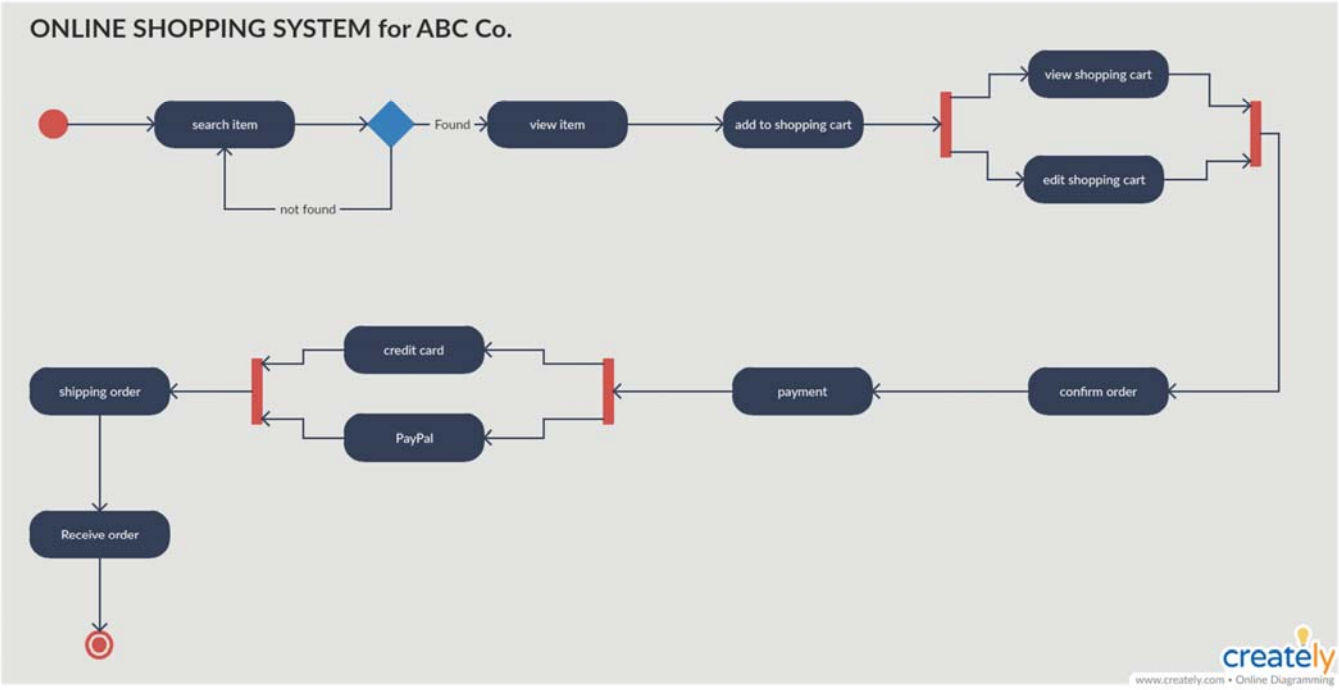
Sequence diagram;
Facebook (FB) user could be authenticated in a web application to allow access to his/her FB resources. Facebook uses **OAuth 2.0** protocol framework which enables web application (called "client"), which is usually not the FB resource owner but is acting on the FB user's behalf, to request access to resources controlled by the FB user and hosted by the FB server. Instead of using the FB user credentials to access protected resources, the web application obtains an access token.



Tampereen yliopisto
Tampere University

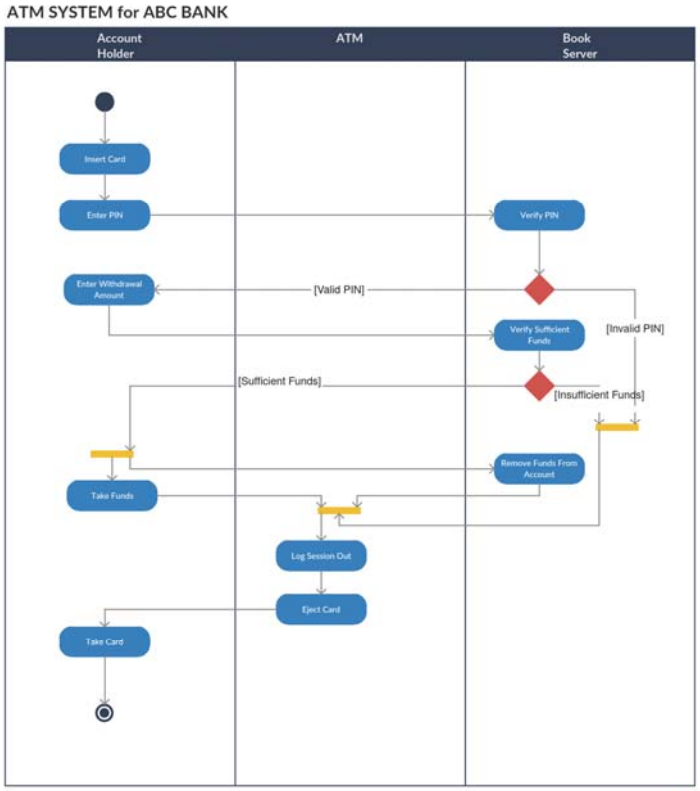
Activity diagrams

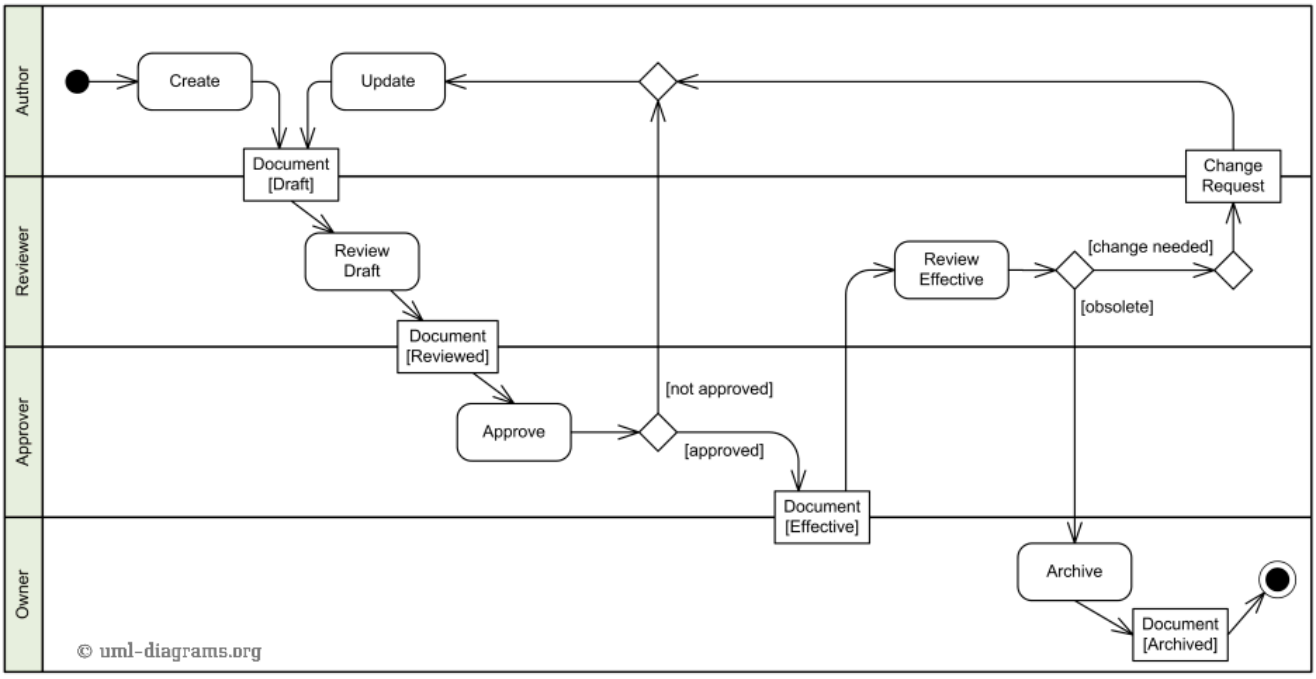
Activity diagram



Activity diagram

Lanes may be vertical or horisontal.





"Swimlane"; this **activity diagram** shows responsibilities of different roles and flow or sequence of document changes. Alternative type of diagram - state machine diagram - could also be used in this case to show how document changes its state over time.

Example of three diagrams

Tampereen yliopisto
Tampere University

student administration system of a university, Use Case diagram

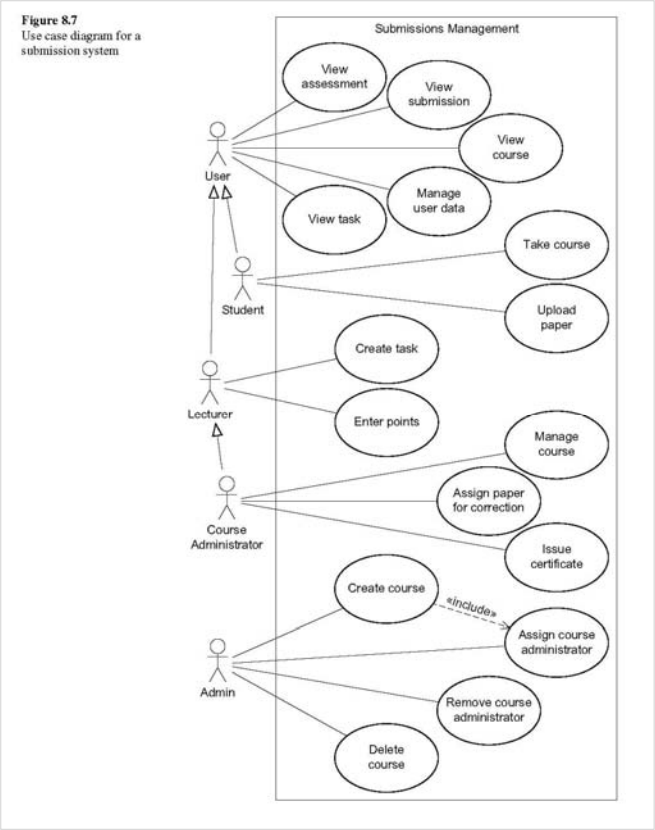


Figure 8.7
Use case diagram for a submission system

[UML@Classroom, 2015]

Tampereen yliopisto
Tampere University

student administration system of a university, Class diagram

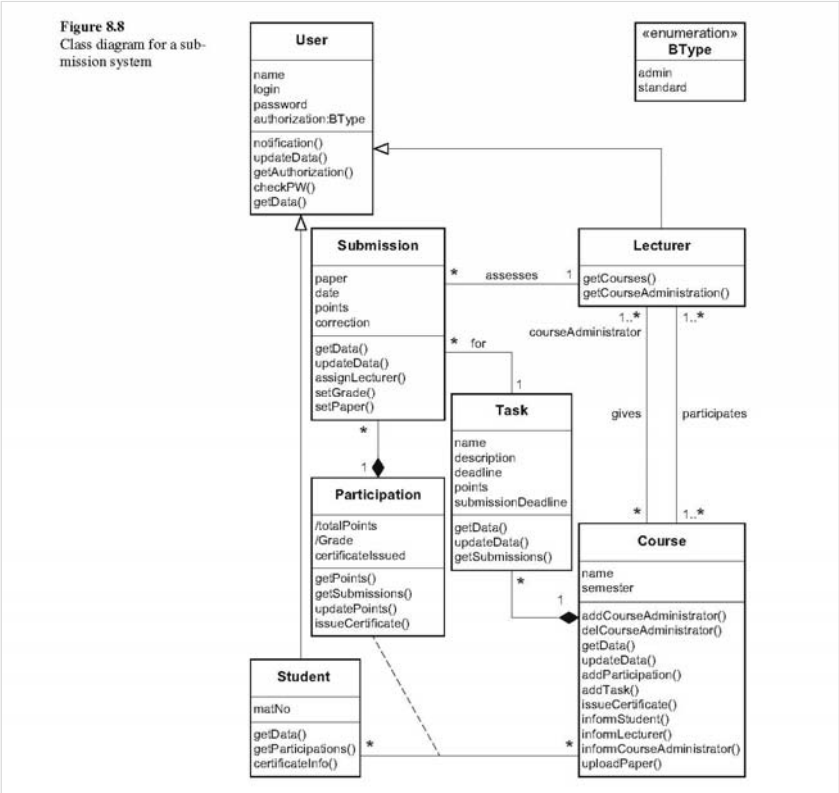
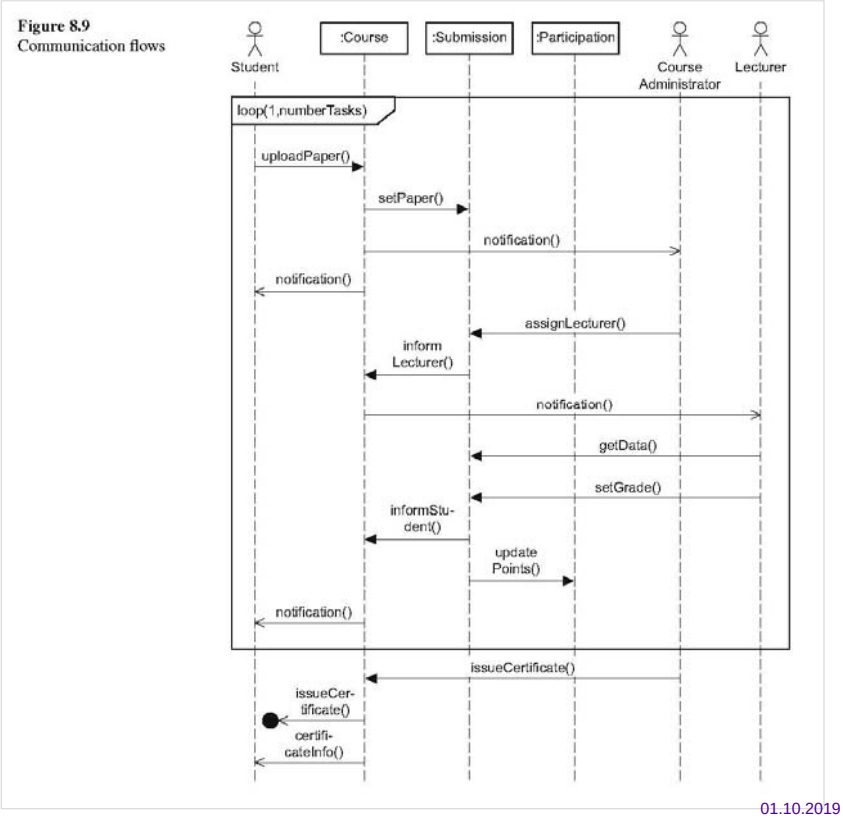


Figure 8.8
Class diagram for a submission system

[UML@Classroom, 2015]

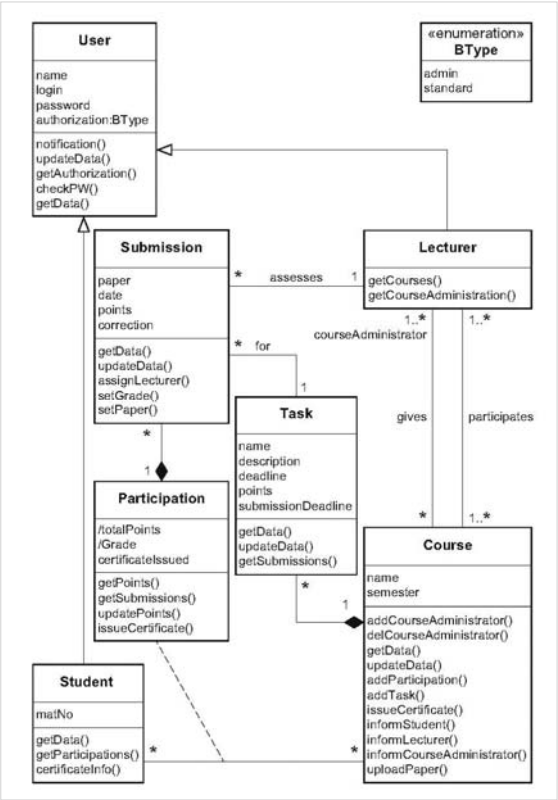
Tampereen yliopisto
Tampere University

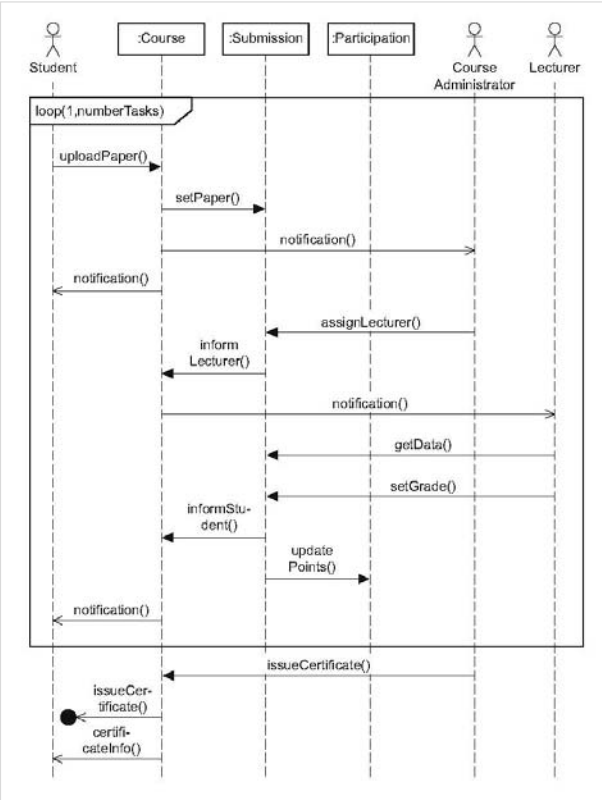
student administration system of a university, Sequence diagram



[UML@Classroom, 2015]

Tampereen yliopisto
Tampere University





[UML@Classroom, 2015]

Tampereen yliopisto
Tampere University

Highlights - What to remember

- state transitions are one important diagram in software development, it **defines in which order actions may/should happen**
- there are several different diagrams (techniques, styles), and terminology varies also
- the main point is to understand (read) different kind of state machines
- there are many many more diagrams and hundreds of details for special cases at UML specification
- remember the availability of Quick Reference Cards/Guides and ”Cheat Sheets”.